Interest Calculator

In this homework assignment, you will be writing a basic interest calculator. This homework covers console input/output and basic arithmetic in Python.

Fixed Interest Calculator

You're going to write a small program that prompts the user to enter a principal amount (the starting amount), followed by the number of years, and then compute the total balance based on annually compounded interest. Don't worry I'll give you the interest formula. For now, we'll just use a **fixed rate of 5%**, but we'll change this later.

1. Create a new Python file called **hw2.py. Your submission must be named hw2.py**. Follow the same process as you did in the first assignment; if you need a refresher, please refer back to Homework 1.

2. We're going to need to import the `math` module to complete this assignment. So put the following at the top of your file:

   ```
   import math
   ```

3. Remember how we used `print()` to write text output to the screen? Python also provides `input()`, which lets us read text (a string) that the user types at the keyboard. First, we want to ask the user to enter the principal amount:

   ```
   principal_str = input("Please enter the principal: ")
   ```

   Remember: at this point `principal_str` is a string not a number. We'll need to convert it into **a floating-point number** before we can use it for the calculations in the next steps. If you've forgotten how to do this look through the Chapter 1 slides.

4. Now repeat the process from step 3, asking the user for the number of years. Use `input()` to print an appropriate prompt and to read a value into a variable. The number of years should be converted into an **integer**.

5. Now that you have your input values, create a variable named `balance` that will hold the final balance. Set `balance` to be equal to an expression that computes the balance based on the original principal and number of years, using 5% as the interest rate compounded annually.

   The mathematical expression for compounded interest is $B = P * (1 + r)^t$, where B is the final balance, P is the original principal, and t is the number of years, and r is the interest rate.

   The exponent should be computed with the `math.pow()` function or `**`. **Note: The interest rate should be between 0 and 1; for example, 5% is 0.05.**

6. Print the final balance using `print()`. Don't just print the number – print a meaningful message like "The final balance is " followed by the amount. Remember that you can chain strings and other values together by using a comma, so:

   ```
   print("The final balance is ", balance)
   ```

   Would print the balance after that message.

7. Run your program, and make sure it works as expected. You can test it with the following values: principal = 1000, number of years = 4. Remember: when you enter numbers larger than 1000, **do not type commas.** The final balance should be roughly 1215.51, or close to it. There will be more than 2 digits after the decimal point, but I wouldn't worry about them.

Changing the interest rate

Now lets modify the program slightly before we finish up, so that it asks the user for the interest rate as well. This makes the program more general and flexible.

1. **After the part of your program where you read the number of years** from the user, add another `input()` prompt to ask the user for the interest rate. We want the user to enter the user to enter the number as an **integer** percentage, without the percent sign and not as a decimal; in other words, for 5%, you would enter "5", **not** "0.05" or "5%". You'll need to convert this value into an integer like in Part I. I would store the value in a variable called `rate` or something similar.
2. Now, modify the formula that computes the balance to use the new `rate` variable instead of the fixed rate you used in Part I. In other words, change your `0.05*t` to read `(rate / 100)*t.`
3. Re-run the program. Try it with the same values as before: principal = 1000, number of years = 4, rate = 5. Did you get the same result? If so, try it with another set of values just to be sure: principal = 7500, number of years = 10, rate = 9. The balance should roughly equal 17755.2.

What to Submit

For this assignment you should submit your **hw2.py** file.

This assignment will be graded automatically. Test your programs thoroughly before submitting them. Make sure that your programs produce correct results for every logically valid test case you can think of. Do not waste submissions on untested code, or on code that does not run with the supplied code from the course website.

Web-CAT will assign a score based on runtime testing of your submission; your best score will be counted; the TAs will later verify that your best submission meets the stated restrictions, and assess penalties if not.

To submit this assignment:

1. Visit http://web-cat.cs.vt.edu in your web browser.
2. Enter your Virginia Tech PID and password in the appropriate fields on the log-in screen, and make sure that **Virginia Tech** is selected as the institution. Click **Login**.
3. The Web-CAT home screen will display useful announcements and assignments that are currently accepting submissions. Find the assignment that you want to submit in the table, and click the "Submit" button next to it.
4. Click the **Browse...** button and select the file you want to upload. The homework assignments and programming projects for this course should be self-contained in a single **.py** file, so you can simply select that one file.
5. Click the **Upload Submission** button. The next page will ask you to review your selection to ensure that you have chosen the right file. If everything looks correct, click **Confirm**.

The next page will show that your assignment is currently queued for grading, with an estimated wait time. This page will refresh itself automatically, and when grading is complete you will be taken to a page with your results.

Pledge

Each of your program submissions must be pledged to conform to the Honor Code requirements for this course. Specifically, you **must** include the following pledge statement in the submitted file:

```
#  <include a description of the purpose of this file/project/package>
#
#  @author <name and surname> (your VT PID)
#  @date   <the date>
#
# Virginia Tech Honor Code Pledge
#  On my honor:
#
#  - I have not discussed the Python language code in my program with
#    anyone other than my instructor or the teaching assistants
#    assigned to this course.
#  - I have not used Python language code obtained from another student,
#    or any other unauthorized source, either modified or unmodified.
# - If any Python language code or documentation used in my program
#    was obtained from another source, such as a text book of coarse
#    notes, that has been clearly noted with a proper citation in
#    the comments of my program.
#  - I have not designed this program in such a way as to defeat or
#    interfere with the normal operation of the Web-Cat Server.
#
#  <your name>
```

Failure to include this pledge in a submission will result in the submission being disallowed during code review.